# Analysis and performance evaluation of some hyperchannel access protocols

O. Spaniol

IRIA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105 78150 Le Chesnay
France
Tél.:954 90 20

Rapports de Recherche

N° 20

# ANALYSIS AND PERFORMANCE EVALUATION OF SOME HYPERCHANNEL ACCESS PROTOCOLS

Otto SPANIOL

Juillet 1980

# ANALYSIS AND PERFORMANCE EVALUATION OF SOME HYPERCHANNEL ACCESS PROTOCOLS

by

Otto SPANIOL *
Institut fuer Informatik
Universitaet Bonn
Wegelerstrasse 6
D - 5300 BONN
(West Germany)

---

* This work was done while the author was at INRIA-Paris.

Résumé

Hyperchannel est un réseau local à priorités fixes pour décider de l'accès au canal. Les performances de ce protocole ont été examinés dans quelques papiers. D'autres études ont été faites pour des protocoles d'accès similaires (BRAM, ...).
Cet article présente une analyse plus détaillée ainsi que l'evaluation des performances du protocole Hyperchannel. Des resultats analytiques ont été obtenus pour les grandeurs les plus importantes du système (débit, temps d'attente, ....) sous différentes charges.
Le système à priorités fixes lèse les clients à basse priorité si le réseau est très chargé. Pour éviter cela, nous proposons une modification du protocole d'accès qui borne le temps d'attente maximum, la moyenne étant alors la même pour chaque client.
Les performance de ce protocole ('Fair Hyperchannel') sont évaluées analytiquement.

Abstract

Hyperchannel is a local network configuration which uses a fixed priority scheme to schedule the access to the global transmission medium. The performance of this protocol has been discussed in a few papers. Several other studies are concerned with similar access protocols, e.g. BRAM. The analytical results which are found in these articles are mostly based on very simplistic assumptions.
This paper presents a much more detailed analysis and performance evaluation of such access schemes. Analytical results are obtained for the most important system parameters such as throughput and waiting times for different load characteristics.
The fixed priority scheme is not very convenient for low priority customers if the network is heavily loaded. To solve this problem we propose a simple modification of the access protocol ("fair Hyperchannel") which guarantees a finite maximum waiting time for all customers as well as the same average waiting time per packet for all customers. The performance of this protocol version is also evaluated by analytical methods.

## 1. THE HYPERCHANNEL ACCESS PROTOCOL (LEVEL 1 PROTOCOL)

Hyperchannel is a local network configuration with with a common transport medium and with a decentralized Ethernet-like access scheme. (see [1] , [4] ); unlike Ethernet, however, access to the common transmission medium after message passing is scheduled by a fixed priority scheme which limits the maximum number of conflicts in the absence of hardware failures. The characteristics of the Hyperchannel access protocol are described in the following:

<u>Hyperchannel characteristics</u>

- fixed number, L, of users $U_1, U_2, \dots, U_L$ ;
- fixed priority scheme: prior $(U_i) >$ prior$(U_{i+1})$ $(i=1,\dots,L-1)$ with no loss of generality ;
- automatic ACK generation and transmission after the correct reception of a message by the receiving station. ACKs have highest priority, thus acknowledgements may be considered as being produced by a fictitious customer $U_0$ ;
- Carrier sensing principle: before starting a transmission, a station senses the bus; whenever the bus is occupied (message passing) the station has to wait until the bus becomes free ;
- conflict-free priority scheduling after message passing: when the bus becomes free after a message passing, access is given to the highest priority station $U_j$ $(j \in \{0,\dots,L\})$ which wants to transmit an ACK or a new message. If there is no station waiting for transmission the system enters the contention mode (random access); as in Ethernet, conflicts are possible in this state due to quasisimultaneous transmissions of at least two stations.

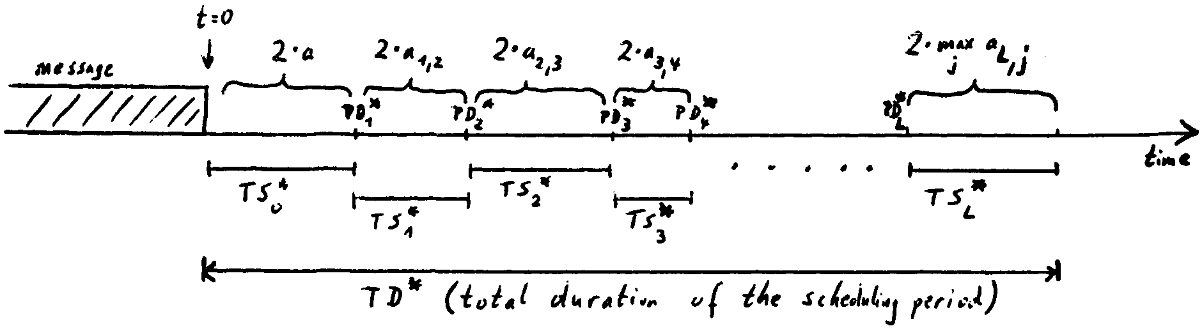
The priority scheme after messagepassing is defined as follows:

Let $a_{i,k}$ = propagation delay between between $U_i$ and $U_k$ $(i,k = 1,\dots,L)$

$$a := \max_{i,k} a_{i,k}$$

(For local networks the maximum propagation delay, a, between two stations is much smaller than the average length of a message).

After a message passing on the channel, a fixed time slot is reserved for the users $U_0,\dots,U_L$ in the following way (see figure 1):




$TS_i^*$ := Time slot for user $U_i$ $(i = 0,\ldots,L)$

$PD_i^*$ := 'Priority Delay' of $U_i$ (beginning of the reserved time slot)

Figure 1: Access scheduling after message passing

The timing intervals are explained by the following reasoning:

After $2a$ seconds it will be certain that no more ACK will be transmitted (since ACKs are generated and transmitted immediately after the correct reception of a message); thus at time $PD_1^* = 2a$ station $U_1$ is allowed to begin a transmission. If $U_1$ doesnot use its access right (this will be remarked by $U_2$ after at most $2\cdot a_{1,2}$ seconds) $U_2$ is allowed to start its transmission at time $PD_2^*$ and so on.
If none of the stations uses its priority delay signal (no transmission starts within the scheduling interval $TD^* = TS_0^* + \ldots + TS_L^*$) the system will turn over to the contention mode.
To realize this priority scheme a timer is used in each of the stations $U_1,\ldots,U_L$ which is initially loaded with the total delay value $TD^*$; the timer of station $U_i$ begins to run down when the channel is sensed idle (i.e. after a message passing) and it indicates the following events:

1. $PD_1^*$ (fixed delay for acknowledgements)
2. $PD_i^*$ (priority delay: access right for $U_i$)
3. $TD^*$ (total delay: beginning of the contention mode)

The timer is stopped whenever it runs down to zero (i.e. at the beginning of the contention mode); it is reset to its initial value whenever the channel is sensed to be busy. Thus the timer of station $U_j$ $(j \geq 2)$ may be reset indefinitely often without signalling $PD_j^*$ due to new messages of higher priority users; for this reason, the maximum waiting time is not upper bounded for these users.
Due to the propagation delays the state of the timers in different stations may be different; this has already been taken into account in the length of the time slots $TS_i^*$.

The time slots $TS_i^*$ are the smallest ones which guarantee that after a message passing one of the waiting stations has a conflict-free access to the channel (in the absence of hardware failures and channel noise); clearly, any greater slot length for user $U_i$ will also result in conflict free scheduling at the expense of a slightly increased scheduling overhead. To simplify the analysis of the protocol we assume that each of the time slots in the scheduling period has the same maximum duration $2 \cdot a$; this modification gives additional safe guards in the scheduling period and results in slightly pessimistic values for the throughput and other system paramaters.

The modified model is described in figure 2:

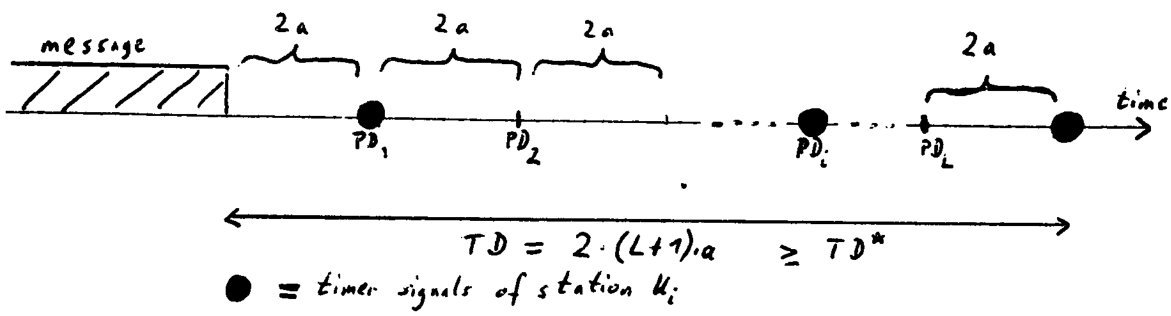



Figure 2: Modification of time slot duration in the scheduling period

## 2. SYSTEM STATES

Four states of the protocols may be distinguished (due to propagation delays different stations see different states during very short time periods, but this does not influence the results of the analysis):

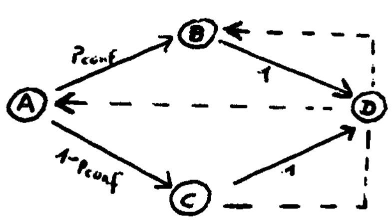
A - contention mode

B - conflict

C - successful transmission (including the ACK which follows)

D - timing (i.e. scheduling period after message passing)

The state transition behaviour of the system is as follows (see figure 3):




$P_{conf}$ := probability that several stations begin a transmission quasisimultaneously.

Figure 3: Transition diagram of the Hyperchannel access protocol

To simplify the analysis it will be assumed that the conflict period in the contention mode has the maximum duration, a, for all stations; this overestimates the probability $p_{conf}$ and results in slightly pessimistic values for the throughput.

The transitions after state D may be interpreted as follows:

' D $\rightarrow$ C ' : Conflict-free scheduling of a waiting customer;

' D $\rightarrow$ A ' : End of the scheduling period; no more customer is waiting;

' D $\rightarrow$ B ' : "Sure Conflict": This transition occurs if and only if:

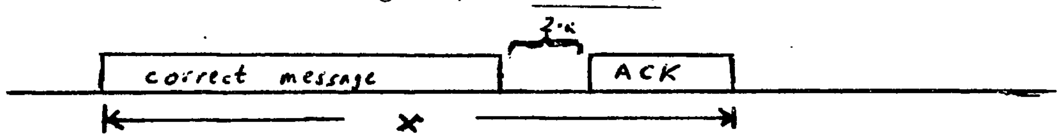
1. No station uses its Priority-Delay-signal in the scheduling period (thus all the timers run down to zero);
2. At least two stations generate a packet after their Priority-Delay-signal but before the Total-Delay-signal. all these stations will start their transmissions when TD is signalled thus a conflict will be produced for sure.

The problem of sure conflicts could be partially solved by randomizing the starting times but this possibility will not be considered for the following reasons:

- randomizing results in some additional channel idle times;
- it will be shown that the probability of a transition 'D $\rightarrow$ B' is very low;
- after a conflict, retransmissions are scheduled conflict-free; thus conflicts do not severely degrade the performance of the system.

In order to be able to analyze the protocol we introduce the following assumptions:

1. The duration, x, of a message which is successfully transmitted (including the acknowledgment) is constant.




2. The packet generating time, $G_k$, of user $U_k$ $(k=1,\ldots,L)$ is exponentially distributed with parameter $\lambda_k$, i.e. $P(G_k \leq h) = 1 - e^{-\lambda_k \cdot h}$ .

Despite these assumptions, the sequence of system states $X_0, X_1, \ldots$ doesnot form a Markow chain since transitions from state D depend on on the number of customers which are waiting for transmission; for example:

$$Pr( X_{n+1}=B \mid X_n =D , X_{n-1}=B ) = 0$$

$$Pr( X_{n+1}=B \mid X_n =D , X_{n-1}=C ) > 0$$

To be more specific, let $X_{n-1} \in \{B,C\}$ and $X_n = D$.

Let $n_1$ := number of customers waiting for transmission at the end of $X_{n-1}$;

$n_2$ := " " " which produce at least one packet in their non-conflict-phase of $X_n$ (i.e. before their PD-signal)

$n_3$ := " " " which produce at least one packet in their conflict period within $X_n$ (i.e.after the PD-signal but before the TD-signal)

The state transitions are determined by the following table:

| $n_1$ | $n_2$ | $n_3$ | $X_{n+1}$ | Comments |
|---|---|---|---|---|
| 1 | % | % | C | Conflict-free scheduling of the waiting customer having the highest priority |
| 0 | 0 | 0 | A | Transition to the contention mode |
| 0 | 0 | 1 | C | Conflict-free scheduling (only one customer starts a transmission at the TD-signal) |
| 0 | 0 | 2 | B | Sure conflict (at least two customers start their transmission at the TD-signal) |
| 0 | 1 | % | C | Conflict-free scheduling (the PD-signal will be used for transmission by exactly one user) |

If $n_1 = 0$ , the transition probabilities are as follows:

<u>Theorem</u>:

Let $p_V^* := \Pr( X_{n+1} = V \mid X_n = D , X_{n-1} = C , n_1 = 0 )$.

If $\lambda_k = \lambda$ $(k = 1,\ldots,L)$ the following values are obtained for $p_V$ :

$$p_A^* = e^{-2aL(L+1)} = 1 - 2aL(L+1) + O(h^2)$$

$$p_B^* = \frac{a^2L^2}{6}(L-1)L(L+1)(3L+2) + O(h^3) = O(h^2)$$

$$p_C^* = 1 - p_A - p_B = 2aL(L+1) + O(h^2)$$

where $$h := \lambda \cdot a \cdot L^2 .$$

The proof of this theorem is given in Appendix (A).

All the probabilities are functions of the parameters $\lambda$, a and L.
By normalizing the message length x to x = 1 the propagation delay, a, will become a very small quantity (for local networks).
The expressions derived in the theorem will be particularly exact if $\lambda \cdot a \cdot L^2 \ll 1$ (i.e. for small propagation delays and/or small user populations).

## 3. PERFORMANCE ANALYSIS

It has been shown that the sequence of states $X_i$ of the Hyperchannel access protocol does not form a Markow chain since transitions from state D to state A, B or C depend on the number of customers waiting for transmission. These transmissions are scheduled conflict-free together with all other packets which arrive during this period. Thus a scheduling period consists of loops between states C and D. We are now going to combine states C and D to a common state E (E = 'Scheduling Period'); see figure 4.

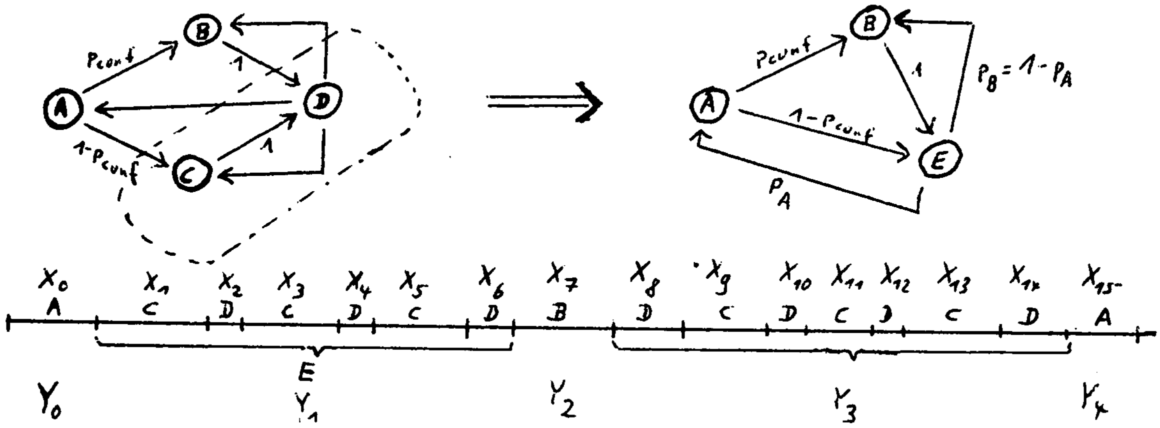



Figure 4 : Reduced state diagram containing scheduling periods as one state

It is easily shown that (under the assumption of exponentially distributed packet generating times) the sequence $Y_0, Y_1, Y_2, \ldots$ of new system states ($Y_i \in \{A,B,E\}$) is a Markow chain.

The state probabilities $\pi_T := \lim_{n\to\infty} \Pr(Y_n = T)$ are easily calculated as:

$$\pi_E = \frac{1}{N} \; ; \quad \pi_A = \frac{1 - p_B}{N} \quad ; \quad \pi_B = \frac{p_{conf} + p_B - p_{conf} \cdot p_B}{N}$$

where $N := 2 + p_{conf}(1-p_B)$ is a normalizing constant.

The calculation of the probabilities $p_{conf}$, $p_A$ and $p_B$ is given in Appendix (B); they are given by:

$$p_{conf} = (L-1)a\lambda - \frac{[(L-1)a\lambda]^2}{2} + \mathcal{O}((a\lambda L))^3$$

$$p_A = 1 - \left(\frac{1}{2} - \frac{2L+1}{3L(L+1)}\right)(a\lambda L(L+1))^2 + \mathcal{O}\left((a\lambda L^2)^3\right) = 1 - \mathcal{O}\left((a\lambda L^2)^2\right)$$

$$p_B = 1 - p_A = \mathcal{O}\left((a\lambda L^2)^2\right)$$

Thus, since $p_B$ is very small for local networks, $\pi_A$ and $\pi_E$ are approximately given by 0.5 . The exact values for these probabilities are given in the following:

$$\left.\begin{aligned} \pi_A &= \tfrac{1}{2} - \tfrac{\alpha}{4} + \tfrac{\kappa^2}{4} - \tfrac{cy^2}{2} \\ \pi_E &= \tfrac{1}{2} - \tfrac{\kappa}{4} + \tfrac{\kappa^2}{4} \\ \pi_B &= \quad \tfrac{\kappa}{2} - \tfrac{\kappa^2}{2} + \tfrac{cy^2}{2} \end{aligned}\right\} + \mathcal{O}\left((a\lambda L^2)^3\right)$$

where $\alpha := (L-1)a\lambda$ ; $y := a\lambda L\cdot(L+1)$ ; $c := \frac{1}{2} - \frac{2L+1}{3L(L+1)}$ .

Let $\delta_i$ := time duration of state $Y_i$ and $T_j := \sum_{r=0}^{j-1} \delta_r$ ; consider now the time-continuous stochastic process $(Z_t \mid t \geq 0)$ which is defined by $Z_t = Y_n$ if $t \in [T_n, T_{n+1})$ .

Then $(Z_t \mid t \geq 0)$ is a Semi-Markow-process (see also [7]) and:

$P_V$ := proportion of time that the process $Z_t$ spends in state V

$$= \lim_{t\to\infty} \Pr(Z_t = V) = \frac{\pi_V \cdot d_V}{\pi_A \cdot d_A + \pi_B \cdot d_B + \pi_E \cdot d_E}$$

where $d_i$ denotes the average duration of state V.

Calculation of the values $d_V$:

$d_A = \frac{1}{\sum_{i=1}^{L} \lambda_i}$ since the idle time (duration of the contention mode) is exponentially distributed with parameter $\sum_{i=1}^{L} \lambda_i$ .

$$d_B = \begin{cases} 2a & \text{if conflicts are detected and transmissions are stopped} \\ x - x_{ACK} & \text{if transmissions are not stopped in case of conflicts} \end{cases}$$

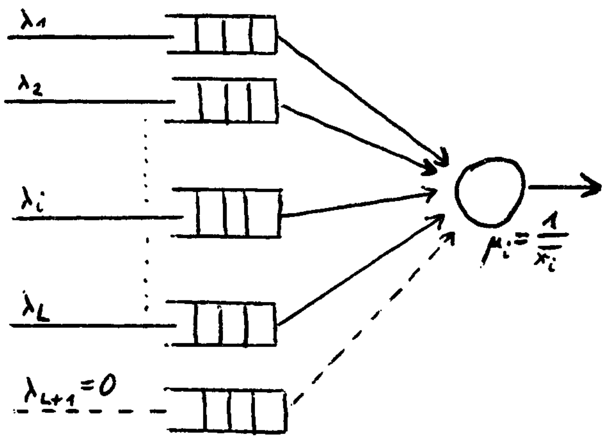
where $x_{ACK}$ is the time duration of the acknowledgment of a correctly received message which has been included in the time x, but after a conflict no acknowledgment will be given.
In the following analysis it will be assumed that conflicting transmissions are not stopped and that $x_{ACK} \ll x$, hence $d_B \approx x$.

$d_E$ : The average duration of state E depends on several factors:

- whether the state preceding E was A or B;
- whether we are concerned with a 'dialogue system' or with a 'file system'
  ( in a file system, $U_i$ may produce new packets even if it has packets waiting for transmission; thus in this case $U_i$ has an unlimited packet waiting queue;
  in dialogue systems, $U_i$ will become blocked after a packet generation until the reception of the ACK which confirms the correct transmission of the packet; hence the packet waiting queue is limited to one place only ).

Several other types of systems could be taken into consideration (packet waiting queues limited to m ($1 < m < \infty$) places); the following analysis, however, is valid for file systems only:

The system may be interpreted as a nonpreemptive priority queueing model consisting of L+1 classes 1,...,L+1 (L=number of users); see figure 5:




$\lambda_i$ $(i=1,\dots,L)$ : packet arrival rate of user $U_i$

$\lambda_{L+1} = 0$ : packet arrival rate of the fictitious user $U_{L+1}$

$\overline{x}_i = x + PD_i = x + 2 \cdot a \cdot i \quad (i=1,\dots,L)$

= service time (including scheduling delays) of packets produced by $U_i$

Figure 5: The nonpreemptive priority queueing model for the analysis of state E

The average duration $d_E$ of state E is now given by:

$d_E = W_{L+1} =$ mean waiting time of a customer of the (fictitious) class L+1 which arrives together with the first customer served during state E.

Thus $d_E$ is determined by the theorem of Cobham (see [3], page 176) which covers the same case with the modification that in our case all the queues are empty at the beginning of state E. Hence:

$$d_E = W_{L+1} = \overline{R} + W'_{L+1} + W''_{L+1}$$

where:

$\overline{R}$ := mean residual service time at the beginning of state E (this is the service time, including scheduling delays, of all the customers which arrive quasisimultaneously at the beginning of state E).

$W'_{L+1}$ := mean service time of the waiting customers in classes 1,...,L+1 which were already present at the beginning of state E

= 0 (since all the queues are empty)

$W''_{L+1}$ := mean service time given to the customers of classes $1,\ldots,L$ which arrive during $W_{L+1}$ (and which are served in the same scheduling period E)

$$= \sum_{i=1}^{L} \rho_i \cdot W_{L+1} \qquad \text{where} \quad \rho_i = \lambda_i \cdot (x + PD_i) = \lambda_i \cdot (x + 2a \cdot i)$$

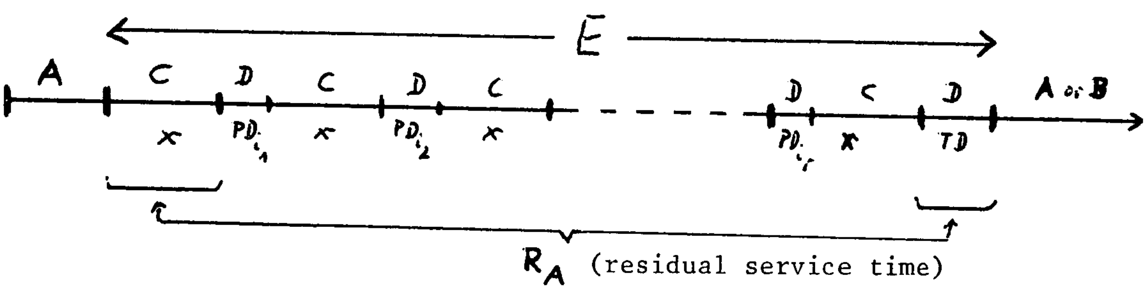
Thus

$$d_E = \begin{cases} \dfrac{\overline{R}}{1 - \beta_L} & \text{if } \beta_L < 1 \\ \infty & \text{if } \beta_L \geq 1 \end{cases}$$

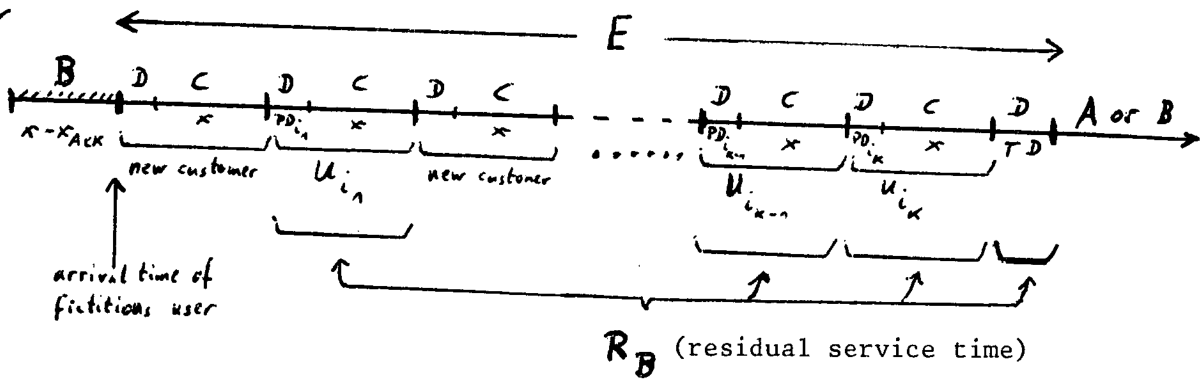
where $\beta_L := \sum_{i=1}^{L} \rho_i$ denotes the total average system load.

If $\beta_L \geq 1$ then the system is overloaded and the high priority customers will monopolize the system at the expense of the low priority customers: If $L^* := \max\{ i \mid \beta_i < 1 \}$ then the mean waiting time until success for customers $U_1,\ldots,U_{L^*}$ is finite whereas the waiting time for customers $U_{L^*+1},\ldots,U_L$ tends to infinity.

The residual service time $\overline{R}$ depends on whether the the state preceding E was A or B (see figure 6).




1. preceding state A: $R_A = x + TD$




2. preceding state B; stations $U_{i_1},\ldots,U_{i_k}$ responsible for the conflict:

$$R_B = \sum_{j=1}^{k} (x + PD_{i_j}) + TD$$

Figure 6: Residual service times in state E (depending on the previous state)

The mean residual service time $\overline{R}$ is now given by:

$$\overline{R} = \overline{R_A} \cdot \Pr(\text{prec. state was A}) + \overline{R_B} \cdot \Pr(\text{prec. state was B})$$

$$= \overline{R_A} \cdot \frac{\pi_A \cdot (1-p_{conf})}{\pi_A \cdot (1-p_{conf}) + \pi_B} + \overline{R_B} \cdot \frac{\pi_B}{\pi_A \cdot (1-p_{conf}) + \pi_B}$$

moreover, $\overline{R_A} = R_A = x + TD$.

If $\lambda_i = \lambda$, the colliding users are equally distributed over $[1:L]$, th $\overline{PD_{i_j}} = a\cdot(L+1)$ which gives the following value for $\overline{R_B}$:

$$\overline{R_B} = \overline{k} \cdot ( x + a\cdot(L+1) ) + TD$$

where $\overline{k}$ is the average number of users which are responsible for a conflict.

Finally:

$$\overline{R} = x + TD + \frac{\pi_B}{\pi_A \cdot (1-p_{conf}) + \pi_B} \cdot \left[ (\overline{k}-1)x + \overline{k}\cdot a \cdot (L+1) \right]$$

$$= x + TD + \left(\alpha - \frac{\alpha^2}{2} + cy^2\right) \cdot \left[ (\overline{k}-1)x + \overline{k}\cdot a (L+1) \right] + \mathcal{O}\left((\lambda a L^2)^3\right)$$

where $\alpha := (L-1)a\lambda$; $y := a\lambda L(L+1)$; $c := \frac{1}{2} - \frac{2L+1}{3L(L+1)}$ (as before).

It may be shown that the probability that at least $i$ customers are participating in a conflict is of the order $\mathcal{O}( (\lambda \cdot a \cdot L)^{i-2})$; thus $\overline{R}$ may be approximated by setting $\overline{k}=3$ and neglecting second order terms; this results in:

$$\overline{R} \approx R^{*} := x + 2a(L+1) + \alpha \cdot ( 2x + 3a(L+1) ).$$

## 4. THROUGHPUT EVALUATION FOR FILE SYSTEMS

Correct messages are transmitted only in state C which belongs to the aggregate state E of the system. Thus the throughput, S, of the Hyperchannel access protocol is given by:

$$S = \frac{\pi_E \cdot d_E \cdot \gamma_C}{\pi_A \cdot d_A + \pi_B \cdot d_B + \pi_E \cdot d_E}$$

where $\gamma_C$ denotes the time proportion of state C within state E.

If state E is composed of exactly m+1 correct transmissions ($0 \leq m \leq \infty$) then the following bounds for $\gamma_C$ are obtained:

1. the state preceding E was A:

$$\frac{x}{x + \frac{1}{2}(d_D + TD)} \leq \gamma_C^{(A)} = \frac{(m+1) \cdot x}{(m+1) \cdot x + m \cdot d_D + TD} \leq \frac{x}{x + d_D}$$

2. the state preceding E was B:

$$\frac{x}{x + d_D + \frac{TD}{2}} \leq \gamma_C^{(B)} = \frac{(m+1)\, x}{(m+1)x + m \cdot d_D + TD} \leq \frac{x}{x + d_D}$$

since $2a \leq d_D \leq 2(L+1)a = TD$.

Thus in both cases the following approximations may be used:

$$\gamma_C \approx \gamma_C^{(A)} \approx \gamma_C^{(B)} \approx \begin{cases} \dfrac{x}{x + d_D} & \text{heavily loaded system} \quad (m \to \infty) \\ \dfrac{x}{x + (d_D + TD)/2} & \text{underloaded system} \quad (m \to 1) \end{cases}$$

We are now calculating the throughput for different system loads:

A. <u>Overloaded system:</u>

The system is overloaded if and only if $\beta_L = L\lambda\,(x + a(L+1)) \geq 1$ (if $\lambda_i = \lambda$ for $i = 1, \ldots, L$).

In this case: $d_E \to \infty$, $\pi_E \to 1$, thus:

$$S \longrightarrow \gamma_C \approx \frac{x}{x + d_D}.$$

Thus the throughput approaches unity if the scheduling time after message passing is much smaller than the length, x, of a message. All the transmissions are scheduled conflict-free but the average waiting time of some low priority users will tend towards infinity.

B. <u>Medium load:</u>

A lower bound for the throughput will be derived.

By inserting the values for the state probabilities, the transition rates and the average residence times we obtain:

$$S = \frac{d_E \cdot \gamma_c}{(1-p_B)\cdot d_A + (p_{conf} + p_B - p_{conf}\cdot p_B)\cdot d_B + d_E} \geq \frac{d_E \cdot \gamma_c}{d_A + p_{conf}\cdot d_B + d_E}$$

$$\geq \frac{x}{x + \frac{1}{2}(\ell_D + TD)} \cdot \frac{1}{1 + \frac{d_A}{d_E} + \frac{p_{conf}\cdot x}{d_E}} \geq \frac{x}{x+TD} \cdot \frac{1}{1 + \frac{(\lambda L)^{-1} + p_{conf}\cdot x}{d_E}}$$

$$\approx \frac{x}{x+TD} \cdot \frac{1}{1 + \frac{(\lambda L)^{-1} - x - a(L+1) + p_{conf}\cdot x}{x + 2(L+1)a + \alpha\cdot(2x + 3a(L+1))}}$$

C. Underloaded system:

This case is characterized by $\beta_L \rightarrow 0$, hence $\lambda L \rightarrow 0$ (if $\lambda_i = \lambda$ )

Furthermore $d_E = \frac{\overline{R}}{1-\beta_L} \rightarrow \frac{x + TD}{1 - L\lambda(x + a(L+1))}$ .

Thus:

$$S \approx \frac{x}{x+TD} \cdot \frac{d_E}{d_E + (\lambda\cdot L)^{-1} + p_{conf}\cdot x}$$

$$= \frac{\lambda L x}{(1-\beta_L)\cdot(1 + \lambda L d_E + \lambda L \cdot p_{conf}\cdot x)}$$

$$= \frac{\lambda L x}{1 - \beta_L + \lambda L (x + TD) + \lambda L\, p_{conf}\, x\,(1-\beta_L)}$$

$$= \frac{\lambda L x}{1 + \lambda L\,(a\cdot(L+1) + p_{conf}\cdot x\cdot(1-\beta_L))}$$

$$\approx \lambda L x \cdot (1 - \lambda L\cdot a\cdot(L+1)) + \mathcal{O}((\lambda L)^3)$$

(the last approximation has been obtained by neglecting higher order terms in $\lambda$ ; thus the term containing $p_{conf}$ has been cancelled since $p_{conf} = (L-1)\cdot a\cdot\lambda + \mathcal{O}((a\lambda L)^2)$.

As expected, the <u>throughput in underloaded systems is almost equal to the total offered load $\lambda\cdot L\cdot x$</u> of the system; this is due to the fact that collisions and scheduling delays are very unlikely under such circumstances.

## 5. THE 'FAIR HYPERCHANNEL' ACCESS PROTOCOL

The protocol version discussed in sections 1 - 4 doesnot guarantee a finite maximum waiting time for low priority customers. In many systems this is not a very serious problem due to the fact that the probability that one customer has to wait very long until success is rather small since in most cases the actual system load is small as compared to the possible transmission rate in local networks; nevertheless there is a considerable interest in protocol versions which guarantee finite maximum waiting times under all circumstances. A rather simple modification of the original access protocol does the job:

Modification of the Hyperchannel access protocol:

During a scheduling period (i.e. during state E of the Hyperchannel access protocol) the customer $U_i$ may transmit at most $m_i$ $(1 \leq m_i < \infty)$ packets.

The vector $M := (m_1,\ldots,m_L)$ determines the behaviour of the modified system. After $m_i$ transmissions user $U_i$ will become blocked until the moment when he receives his TD-signal announcing the end of state E. During any eventual transmission blocking period this user might already prepare new packets for transmission.

Remark: The special choice $M = (1,\ldots,1)$ is very similar to the 'Slotted Ethernet' concept (see [7]); as opposed to this concept, however, in the modified Hyperchannel protocol new customers may generate a packet and contend for channel access even during scheduling periods. Due to this reason, the analytical results of the Sloteed Ethernet system cannot be generalized on the Hyperchannel access protocol.

It is obvious that the modified Hyperchannel access protocol guarantees finite maximum waiting times for all customers since high priority customers cannot monopolize the system any more. Nevertheless, the actual waiting times are different since high priority customers are favoured in that they may capture the transmission medium earlier then low priority customers. To compensate for this discrimination we propose to give more access rights to low priority customers; this principle will be called 'FAIR HYPERCHANNEL' and is defined in the following way:

FAIR HYPERCHANNEL ACCESS PROTOCOL

During a scheduling period, customer $U_i$ may transmit up to $m_i$ packets where $1 \leq m_1 \leq m_2 \leq \ldots\ldots \leq m_L$ and where the values $m_i$ are chosen such that the maximum waiting time per packet transmitted is the same for each of the customers.

To determine the values $m_i$ which are in accordance with this criterion we assume that $m_i \in \mathbb{R}_+$ rather than $m_i \in \mathbb{N}_+$, since in general it will be impossible to satisfy exactly the condition given above if the $m_i$ are integer valued. After the determination of optimal but real valued paramaters $m_i$ we have to approximate them by integer values such that the fairness criterion is satisfied as exactly as possible.

The maximum waiting time for customer $U_i$ is composed of three parts $W_{k,i}^{max}$ $(k = 1,2,3)$:

a. Waiting time due to a possible blocking situation during the scheduling period:
   In the worst case, $U_i$ could have already transmitted $m_i$ packets and will from that moment on be blocked until all other customers $U_j$ finish the transmission of the maximum number, $m_j$, of packets.
   Thus:
   $W_{1,i}^{max} = \sum_{j \neq i} m_j \cdot (x + PD_j) + TD$

b. Waiting time due to a possible conflict after state E (transition $E \rightarrow B$):
   $W_{2,i}^{max} = x$ .

c. Waiting time due to transmissions of prioritized customers:
   In the worst case, all of the customers $U_1, \ldots, U_{i-1}$ might use all of their access rights before the first transmission of user $U_i$:
   $W_{3,i}^{max} = \sum_{j=1}^{i-1} m_j \, (x + PD)$

In the following, timer periods $PD_i$ and TD will be neglected since they are usually very short as compared to transmission times. Furthermore, the real numbers $m_i$ which will result from our calculations have to be replaced by integer numbers; thus we cannot expect that the fairness criterion will be exactly satisfied.

Thus the total maximum waiting time per packet of user $U_i$ is given by:

$$
\begin{aligned}
W_{total}^{(i)} &= ( W_{1,i}^{max} + W_{2,i}^{max} + W_{3,i}^{max} ) / m_i \\
&= \Big( \sum_{j=1}^{L} m_j \cdot x - m_i \cdot x + x + \sum_{j=1}^{i-1} m_j \cdot x \Big) / m_i \\
&= \frac{x}{m_i} \cdot \left[ S_L + 1 - S_{i-1} \right] - x
\end{aligned}
$$

where $S_k := \sum_{i=1}^{k} m_i \quad ( k = 1,\dots,L).$

The fairness criterion is defined by the following equation:

$$W_{total}^{(i)} = W_{total}^{(i+1)} \qquad (i=1,\dots,L-1);$$

this condition results in:

$$\frac{1}{m_i} \cdot \left[ S_L + 1 + S_{i-1} \right] = \frac{1}{m_{i+1}} \cdot \left[ S_L + 1 + S_i \right]$$

thus:

$$m_{i+1} = \frac{m_i \cdot (S_L + 1 + S_i)}{S_L + 1 + S_{i-1}} = m_i + \frac{m_i^2}{S_L + 1 + S_{i-1}}$$

It is easily shown by induction that the $m_i$ are given by:

$$m_k = m_1 \cdot \left( 1 + \frac{m_1}{S_L + 1} \right)^{k-1} \qquad (k = 1,\dots,L)$$

By summing over $m_i$ we obtain another relation which can be used to determine the unknown $S_L$:

$$S_L = \sum_{k=1}^{L} m_k = m_1 \cdot \sum_{k=1}^{L} \left( 1 + \frac{m_1}{S_L + 1} \right)^{k-1} = (S_L + 1) \cdot \left[ \left( 1 + \frac{m_1}{S_L + 1} \right)^{L} - 1 \right]$$

$$\Rightarrow \quad m_1 = (S_L + 1) \cdot \left( \sqrt[L]{\frac{S_L}{S_L + 1} + 1} - 1 \right)$$

This equation cannot be explicitly solved for $S_L$, but if the number, L, of customers is large, then the parameter $S_L$ is also large (due to $m_1 \geq 1$) and the term $\sqrt[L]{\frac{S_L}{S_L+1} + 1}$ may be approximated by $\sqrt[L]{2}$; this approximation leads to:

$$1 + \frac{m_1}{S_L + 1} \lessapprox \sqrt[L]{2}$$

$$\Rightarrow \quad m_k = m_1 \cdot \left( 1 + \frac{m_1}{S_L + 1} \right)^{k-1} \lessapprox m_1 \cdot 2^{\frac{k-1}{L}} \qquad (k = 1,\dots,L).$$

This approximation also determines the range of the paramaters $m_i$ since:

$$m_1 \leq m_i \leq m_L < m_1 \cdot 2^{(L-1)/L} < 2 \cdot m_1$$

Thus the optimal parameters $m_i$ increase very slowly.

We conclude that also the optimal integer valued parameters $m_i^{+}$ have to be choosen from the interval $[m_1 : 2 \cdot m_1]$.

Consider for example the special case $m_1 = 1$.
Then the integer valued fairness parameters $m_i^+$ are determined by:

$$m_i^+ = \begin{cases} 1 & \text{if } i = 1,\dots,h \\ 2 & \text{if } i = h+1,\dots,L \end{cases}$$

where $h$ is determined by the condition $m_h = 1.5$, i.e. $2^{(h-1)/L} = 1.5$

thus $h = L \cdot \log_2 1.5 + 1 \approx 0.58\,L + 1$.

<u>Remark</u>: The maximum total waiting time per packet is given by

$$W_{total}^{(i)} = W_{total}^{(1)} = \frac{x}{m_1} \cdot (S_L + 1) - x.$$

If $m_1 = 1$ this reduces to $W_{total}^{(i)} = x \cdot S_L$.
Thus in this case the maximum waiting time grows proportionally to the maximum number of transmissions during a scheduling period (state E).

## 6. PERFORMANCE ANALYSIS OF 'FAIR HYPERCHANNEL'

The state diagram of 'Fair Hyperchannel' coincides wity the state diagram of the original protocol version (see figure 7).

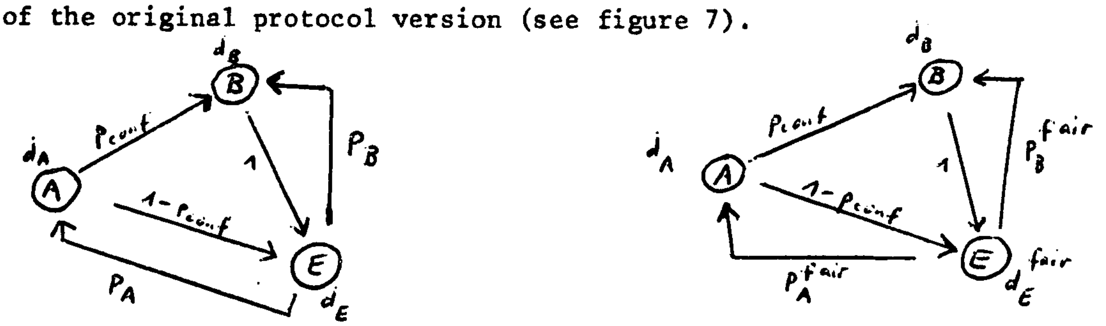



FIGURE 7: 'Original Hyperchannel' versus 'Fair Hyperchannel'

The state transition probabilities as well as the duration of the states, however, are influenced by the modification of the protocol; this influence becomes more and more serious when the load of the network is growing; in underloaded systems blocking situations (i.e. a situation where a customer $U_i$ tries to transmit more than $m_i$ packets during one scheduling period) do not occur very often.

The similarities and the differences of both protocol versions are summarized in the following table:

| Unchanged | Changed |
|---|---|
| 1. $p_{conf}$ | 1. Transition probabilities $p_A^{fair}$, $p_B^{fair}$ |
| 2. $d_A$ , $d_B$ | 2. $d_E^{fair}$ |
| 3. Formulae for $\pi_A, \pi_B, \pi_E$ | 3. Values of $\pi_A, \pi_B, \pi_E$ |
| 4. Formula for the throughput S | 4. Value of throughput $S^{fair}$. |

It will be shown that in heavily loaded systems the probability $p_B^{fair}$ (i.e. the probability of a 'sure conflict') is much higher than the corresponding probability $p_B$; on the other hand, $d_E^{fair}$ will become much smaller than $d_E$ since monopolizing the 'Fair Hyperchannel' system (which results in extremely large values for $d_E$) is impossible. Finally, the throughput $S^{fair}$ will be slightly inferior to S in heavily loaded systems since blocking situations lead to a certain increase of the number of conflicts which results in.a decrease of throughput. As has already been mentioned, the throughput and other system parameters are almost unchanged in moderately loaded systems.

Calculation of $d_E^{fair}$ for file systems:

The average duration of state E depends on the state preceding E:

1. Preceding state = A:

   As in section 3, the system may be interpreted as a nonpreemptive priority queueing model with M+1 customer classes:

   $$d_E^{(A)} = \overline{R^{fair}} + W''_{L+1}$$

   where $\overline{R^{fair}} = \overline{R} = x + TD$ = residual service time of the first user

   and $W''_{L+1} = \sum_{i=1}^{L} \min \{ m_i^{(k)} \cdot (x + PD_i) ; \rho_i \cdot d_E^{(A)} \}$

   = service time of customers which arrive during $d_E^{(A)}$.

   Hereby k denotes the index of the first customer served in state E

   and $$m_i^{(k)} := \begin{cases} m_i & \text{if } k \neq i \\ m_i - 1 & \text{if } k = i \end{cases}$$

   It will be very hard to calculate $d_E^{(A)}$ from this general formula; thus we restrict ourselves on the cases of underloaded as well as of overloaded systems; furthermore $\lambda_i = \lambda$ will be assumed:

a. Underloaded system, i.e. $\lambda \cdot L \ll 1$:

$$d_E^{(A)} \approx x + TD + \sum_{i=1}^{L} (x + PD_i) \cdot \lambda d_E^{(A)} = x + TD + \lambda \cdot d_E^{(A)} \cdot \left(Lx + a L(L+1)\right)$$

$$\Rightarrow d_E^{(A)} \approx \frac{x + TD}{1 - \lambda L (x + a \cdot (L+1))} .$$

b. Overloaded system, i.e. $\lambda \cdot L \gg 1$:

$$d_E^{(A)} \approx x + TD + \sum_{i=1}^{L} (x + PD_i) \cdot m_i^{(k)} = \sum_{i=1}^{L} m_i \cdot (x + PD_i) + TD - PD_k$$

$$\Rightarrow d_E^{(A)} \approx S_L \cdot x + 2a\left(\sum_{i=1}^{L} i \cdot m_i + L + 1 - k\right).$$

2. Preceding state = B:

In this case, the actual queue size of customer $U_i$ depends on the number of access rights, $m_i$, per scheduling period as well as on the fact whether the customer was blocked in the recent past or not. A general evaluation of $d_E^{(A)}$ seems to be very difficult, but for overloaded systems obviously the same result will be obtained as above:

$$d_E^{(A)} \approx d_E^{(B)} \qquad \text{(for overloaded systems).}$$

It should be noted that the probability of being in state B is very small for underloaded systems, thus it is not necessary to evaluate $d_E^{(B)}$ in this case.

Throughput analysis for file systems:

The throughput of the 'Fair Hyperchannel' Protocol is given by the formula:

$$S^{fair} = \frac{\pi_E^{fair} \cdot d_E^{fair} \cdot \gamma_C^{fair}}{\pi_A^{fair} \cdot d_A + \pi_B^{fair} \cdot d_B + \pi_E^{fair} \cdot d_E^{fair}}$$

where (as in section 4) $\gamma_C^{fair}$ is given by:

$$\gamma_C^{fair} \approx \gamma_C \approx \begin{cases} \dfrac{x}{x + d_D} & \text{heavily loaded systems} \\[2ex] \dfrac{x}{x + (d_B + TD)/2} & \text{underloaded systems} \end{cases}$$

We are now going to evaluate $S^{fair}$ for underloaded as well as for overloaded systems:

1. Overloaded systems:

In an overloaded system all stations use their transmission facilities and become blocked due to the limitation of $m_i$; furthermore a conflict will occur for sure after the scheduling period. Hence:

$$p_B^{fair} \rightarrow 1,\ p_A^{fair} \rightarrow 0\ ,\ d_E^{fair} \longrightarrow d_E^{(B)}, \pi_E \rightarrow \frac{1}{2}\ ,\ \pi_B \rightarrow \frac{1}{2}\ ,\ \pi_A \rightarrow 0\ .$$

Thus the throughput $S^{fair}$ is given by:

$$S^{fair} \longrightarrow \frac{d_E^{fair} \cdot \gamma_e^{fair}}{d_B + d_E^{fair}} \approx \frac{x \cdot d_E^{fair}}{(x + d_D) \cdot (x + d_E^{fair})}$$

$$= \frac{1}{(1 + \frac{d_D}{x}) \cdot (1 + \frac{x}{d_E^{fair}})} \geq \frac{1}{(1 + \frac{d_D}{x}) \cdot (1 + \frac{1}{s_L})}$$

since $d_E^{fair} \approx S_L \cdot x + 2a \cdot \left(\sum_{i=1}^{L} i \cdot m_i + L + 1 - k\right) \geq S_L \cdot x$ .

2. Underloaded systems:

   In underloaded systems, the throughput is not degraded due to blocking situations. The parameters coincide (up to terms of higher order) with the corresponding parameters of the original protocol version. Thus (see section 4): .

$$S^{fair} \approx S \approx \lambda L x \cdot \left(1 - \lambda L a \cdot (L+1)\right) + \mathcal{O}\left((\lambda L)^3\right).$$

## 7. COMPARISON WITH OTHER LOCAL NETWORK ACCESS SCHEMES

A very complete discussion of different access schemes has been presented by Eldessouky (see [5]).
Here we intend to compare the two versions of Hyperchannel access protocols ('Original' and 'Fair' Hyperchannel) with three similar schemes:

1. Slotted Ethernet [7]:

   It has already been mentioned that Slotted Ethernet shares many features with 'Fair Hyperchannel'. In the following, a list of similarities as well as of differences will be presented:

   Similarities:

   a. Both schemes contain 'scheduling periods'. Retransmissions are scheduled conflictfree due to priority rules.
   b. The 'reservation slot' TD depends on the (fixed) number of users. As opposed to Slotted Ethernet, however, this 'slot' is partially used for message transmissions in the Hyperchannel access protocols.
   c. At most one conflict per packet is possible.
   d. Maximum waiting times are upper bounded for Slotted Ethernet as well as for 'Fair Hyperchannel'.
   e. High throughput for overloaded systems, few collisions in underloaded systems.

Differences:

a. Priority rules in Hyperchannel are fixed, but they are arbitrary in Slotted Ethernet.

b. Hyperchannel access is possible for variable length messages, too.

c. In 'Original Hyperchannel', the channel may be monopolized by high priority users.

d. In Slotted Ethernet, conflicting transmissions have to be stopped whereas in the Hyperchannel schemes the immediate detection of conflicts is not necessary.

2. BRAM

Another scheme of considerable interest which is similar to the Hyperchannel access protocols is the so called BRAM (Broadcast Recognizing Access Method); see [2]).
Four versions of this methods have been discussed in [2]:

| | |
|---|---|
| a. Prioritized BRAM | (comparable to 'Original Hyperchannel') |
| b. Fair BRAM | ( " " 'Fair Hyperchannel') |
| c. Parametric prioritized BRAM | |
| d. Parametric fair BRAM | |

The parametric versions have been introduced to shorten the timing periods (when the user population is large) by giving the same timing delays to several customers. In these scheme, which could also be defined for our Hyperchannel access methods, repeated conflicts of members of one group are possible; hence the number of collisions per packet is not upper bounded. These methodes combine advantages and disadvantages of both the Hyperchannel and the ALOHA (i.e. random access) protocols.
The main differences between the nonparametric BRAM versions and the Hyperchannel access protocols are the following ones:
Access rights are given in BRAM to the customers on a cyclic basis (in prioritized BRAM, furthermore, the actually transmitting customer may monopolize the channel). To realize this access scheme, any customer has to recognize the identification of the user which is actually transmitting; the decision whether he is allowed to transmit or not depend on this 'actual priority'. Hyperchannel access is much simpler in that users are not forced to detect the authorship of ongoing transmissions. Moreover, fairness is realized in 'Fair Hyperchannel' by a very simple stopping rule; in 'Fair BRAM', a similar effect has been obtained by a modification of priority cycling rules. Finally, in 'Fair Hyperchannel'

$m_i$ packets may be transmitted per cycle; 'Fair BRAM' is much less flexible since it allows at most one transmission per cycle.

3. Algorithms (A1) and (A2)

A very interesting scheme has been published by Hamacher and Shedler [6]. They discuss two conflictfree channel access algorithms where conflicts are avoided by means of an additional control link. The first algorithm is unfair in the sense that the channel may be monopolized by high priority users, the second one (A2) circumvents this monopolizing possibility at the expense of slightly increased overhead which reduces the throughput especially in the case of underloaded systems.
It is easy to see that (A1) and (A2) are strongly related to 'Original' and to 'Fair' Hyperchannel respectively. It should be noted that (A1) has been analyzed by similar methods (Semi Markow chain techniques) as have been used for the performance evaluation of Hyperchannel systems in this paper.

## 8. CONCLUSION

The Hyperchannel access protocol (level 1 protocol) has been described by means of state diagrams. State transition probabilities and average residence times have been evaluated; following that, throughput and waiting time formulae have been derived in terms of the following system parameters: a (propagation delay in the network), L (number of users) and $\lambda$ (input rate of messages per user). Different types of user behaviour ('dialogue' and 'file' systems) have been discussed.

The fixed priority access method after message passing has the obvious disadvantage that the channel may be monopolized by high priority users; to solve this problem, a modification has been proposed which is very easy to implement: in this scheme, user $U_i$ is forced to stop contending for transmission if he transmitted $m_i$ packets without seeing the states 'conflict' or 'contention mode'. The values $m_i$ may be chosen such that the maximum possible waiting time per packet is the same for all customers; this scheme is called 'Fair Hyperchannel' and has been evaluated by the same techniques which have been applied for the original Hyperchannel protocol. The results obtained for 'Original Hyperchannel' and for 'Fair Hyperchannel' are compared with each other as well as with other local network access protocols.

APPENDIX (A): Calculation of transition probabilitites $p_A^*$, $p_B^*$, $p_C^*$

Let $p_V^* := \Pr( X_{n+1} = V \mid X_n = D , X_{n-1} = C , n_1 = 0 );$

if $G_j$ denotes the (exponentially distributed) packet generating time of user $U_j$, then we define:

$$r_i^0 := \Pr( G_i \geq TD ) = e^{-\lambda_i \cdot TD}$$

$$r_i^1 := \Pr( PD_i \leq G_i < TD ) = e^{-\lambda_i \cdot PD_i} - e^{-\lambda_i \cdot TD}$$

$$r_i^2 := \Pr( G_i < PD_i ) = 1 - e^{-\lambda_i \cdot PD_i} .$$

Using these abbreviations we now obtain the probabilities $p_V^*$ as follows:

$$\begin{aligned} p_A^* &= \Pr( \text{no packets are generated during } X_n \mid n_1 = 0 ) \\ &= r_1^0 \cdot \ldots \cdot r_L^0 = e^{-\sum \lambda_i \cdot TD} \overset{(\lambda_i = \lambda)}{=} e^{-\lambda \cdot L \cdot TD} = e^{-2aL(L+1)} \\ &= 1 - 2aL(L+1) + \mathcal{O}(h^2) \quad \text{where } h := \lambda \cdot a \cdot L^2 . \end{aligned}$$

Similarly:

$$\begin{aligned} p_B^* &= \Pr( \text{at least two packets are generated in the conflict-phase,} \\ &\qquad \text{but no packet generation in the non-conflict-phase} \mid n_1 = 0 ) \\ &= \sum_{j=2}^{L} \Pr( j \text{ packets in conflict-phase; 0 packets in non-conflict-phase}) \\ &= \sum_{\{(j_1,\ldots,j_L) \mid j_r \in \{0,1\},\ \sum j_k \geq 2\}} r_1^{j_1} \cdot \ldots \cdot r_L^{j_L} \\ &= \sum_{\{(j_1,\ldots,j_L) \mid j_r \in \{0,1\}\}} r_1^{j_1} \cdot \ldots \cdot r_L^{j_L} - \sum_{\{(j_1,\ldots,j_L) \mid \sum j_r \leq 1\}} r_1^{j_1} \cdot \ldots \cdot r_L^{j_L} \\ &= (1-r_1^2) \cdot \ldots \cdot (1-r_L^2) - r_1^0 \cdot \ldots \cdot r_L^0 - r_1^1 \cdot r_2^0 \cdot \ldots \cdot r_L^0 \\ &\qquad - \ldots\ldots - r_1^0 \cdot \ldots \cdot r_{L-1}^0 \cdot r_L^1 \end{aligned}$$

By inserting the values for $r_i^k$ this reduces to:

$$p_B^* = e^{-a\lambda L(L+1)} - e^{-2a\lambda L(L+1)} - e^{-2a\lambda (L-1)(L+1)} \cdot \sum_{i=1}^{L} \left( e^{-2a\lambda i} - e^{-2a\lambda (L+1)} \right)$$

Since the propagation delay, a, is a very small quantity, we approximate $e^{-ca}$ by $1 - ca + \frac{(ca)^2}{2} + \mathcal{O}((ca)^3)$ and obtain after simple algebraic manipulations:

$$p_B^* = \frac{1}{6} a^2 \lambda^2 (L-1) L (L+1)(3L+2) + \mathcal{O}\left( (a\lambda L^2)^3 \right) .$$

APPENDIX (B): Calculation of the transitionprobabilities $p_A$, $p_B$ and $p_{conf}$:

Using the transition table and the abbreviations of sections 2 and 3 we obtain:

$$p_A = \Pr(\text{ 0 packets are generated in the conflict-phase } \mid n_1=0,\ n_2=0,\ n_3 \neq 1\ )$$

(if exactly one packet would be produced in the conflict-phase, but no packet in the non-conflict-phase, i.e. $n_2=0$, $n_3=1$, then the next state to be visited is C, thus state E would not end in this case).

Hence:

$$p_A = \frac{\Pr(\text{ 0 packet generations in the conflict-phase})}{1 - \Pr(\text{ 1 packet generation in the conflict-phase})} \stackrel{\text{def}}{=} \frac{Q}{N}$$

Now $$Q = \Pr(G_i > TD - PD_i) \stackrel{(\lambda_i=\lambda)}{=} \prod_{i=1}^{L} e^{-2(L+1-i)\lambda a} = e^{a\cdot\lambda L(L+1)}.$$

$$\begin{aligned} N &= 1 - \sum_{i=1}^{L} \Pr(G_i \le TD-PD_i\ ;\ G_j > TD-PD_j\ (j \neq i)) \\ &= 1 - e^{-a\lambda L(L+1)} \cdot \sum_{i=1}^{L}\left(e^{2(L+1-i)a\lambda} - 1\right) \\ &= 1 - e^{-a\lambda L(L+1)} \cdot \sum_{i=1}^{L}\left(e^{2ai\lambda} - 1\right) \end{aligned}$$

Thus:

$$\begin{aligned} p_A &= \frac{1-y+\frac{y^2}{2}}{1-\left(1-y+\frac{y^2}{2}\right)\cdot\sum_{i=1}^{L}\left(1+2ia\lambda+\frac{(2ia\lambda)^2}{2}-1\right)} + \mathcal{O}\left((a\lambda L^2)\right)^3 \\ &= \frac{1-y+\frac{y^2}{2}}{1-y-y^2\cdot\frac{2L+1}{3(L+1)L}+y^2} + \mathcal{O}\left((a\lambda L^2)\right)^3 \\ &= 1-\left(\frac{1}{2}-\frac{2L+1}{3L(L+1)}\right)y^2 + \mathcal{O}\left((a\lambda L^2)\right)^3 \end{aligned}$$

where $y := a\cdot\lambda\cdot L(L+1)$.

$$p_B = 1 - p_A = \left(\frac{1}{2} - \frac{2L+1}{3L(L+1)}\right)y^2 + \mathcal{O}\left((a\lambda L^2)\right)^3 .$$

$p_{conf}$ is the probability that at least two customers start a transmission quasisimultaneously (i.e. within a propagation delay period) in the contention mode; let $U_k$ be the customer who produces the first packet in this mode. Then:

$$\begin{aligned} p_{conf} &= 1 - \Pr(G_j > a\ (j \neq k)) = 1 - (e^{-a\lambda})^{L-1} \\ &= (L-1)a - \frac{((L-1)a)^2}{2} + \mathcal{O}((a\lambda L)^3) \end{aligned}$$

REFERENCES

[1] I. Chlamtac, W.R. Franta: Message-based priority access to Local Networks; Computer Communications 3 (1980) 77-83.

[2] I. Chlamtac, W.R. Franta, K.D. Levin: BRAM, the Broadcast Recognizing Access Method; IEEE Transactions on Communications COM-27 (1979) 1183-1189.

[3] E.G. Coffman, P. Denning: Operating Systems Theory. Prentice-Hall 1973.

[4] J.E. Donnelley, J.W. Yey: Interaction Between Protocol Levels in a Prioritized CSMA Broadcast Network. Computer Networks 3 (1979) 9-23.

[5] M. Eldessouky: Modélisation et Evaluation des Performances des Protocoles d'Accès. Thèse de Doctorat, Université de Limoges, 1980.

[6] V.C. Hamacher, G.S. Shedler: Performance of a Collision-free Local Bus Network having Asynchronous Distributed Control. IBM Resaerch Report RJ 2624 (33802), 1979.

[7] O. Spaniol: Modelling of Local Computer Networks. Computer Networks 3 (1979) 315-326.